17.5/25

KING SAUD UNIVERSITY

COLLEGE OF COMPUTER & INFORMATION SCIENCES

DEPT OF COMPUTER SCIENCE

CSC311 Computer Algorithms
Third Semester 1444 AH (Spring 2023)
Mid-term Examination: Wed 10.05.2023 C.E. (duration = 1:30 hours)
Instructors: Prof Aqil M. Azmi + Prof M. Maher Ben Ismail

Name: [redacted] **ID:** [redacted]

1. **[Marks 5]** 3.5/5
Give a suitable big-O, big-Ω, small-o, and small-w for each of the following (n is the size of the input problem),

		Big-O	Big-Ω	small-o	small-w
(a)	$\log n + 10000$	$O(\log n)$ ✓	$\Omega(\log n)$ ✓	$o(n)$ ✓	$w(1)$ ✓
(b)	$n \log n + 15n + 0.002n^2$	$O(n \log n)$ ✗	$\Omega(n \log n)$ ✗	~~$n \log n$~~ $o(n^2)$ ✗	$w(\log n)$ ✓
(c)	$37n + \sqrt{n}$	$O(n)$ ✓	$\Omega(n)$ ✓	$o(\frac{\cancel{\log n}}{n^2})$ ✓	$w(\log n)$ ✓
(d)	$1000n^2 + 17n + 2^n$	$O(n^2)$ ✗	$\Omega(n^2)$ ✗	$o(n^3)$ ✗	$w(n)$ ✓
(e)	$2^{10} + 3^5$	$O(1)$ ✓	$\Omega(1)$ ✓	$o(\frac{[illegible]}{\log n})$ ✓	$w(1)$ ✓

2. **[Marks 3]** 1/3
What is the value of C and n_0 such that $0.01n \log n + 200n + 6 \in \Omega(n \log n)$.

~~0,01 n log n + 200n + 6 ≥ n log n~~

$0{,}01\, n \log n + 200n + 6 \geq n \log n$

$c = 1$ ✗ , $n_0 = \dfrac{\sum_{i=0}^{k-1} a_k}{a_k - c} = \dfrac{206{,}01}{0{,}01 - 1} = 208$

Scanned with CamScanner

136358

$n=8$

$\sum_{i=1}^{5/3} i$

$3/i$

$\sum_{i=1}^{N/3} 5+(3(i-1))$

~~x = 5 i=1~~
~~x = 8 i=2~~
~~x = 11 i=3~~

$n=6$
$i=1 \rightarrow x=5$
$i=2 \rightarrow x=5$
$i=3 \rightarrow x=5+3$
$i=4 \rightarrow x=5+3$
$i=5 \rightarrow x=5+3$
$i=6 \rightarrow x=5+3+6=14$

4.5/5

3. **[Marks 5]**
Consider the code fragment,

```
x ← 5
for i ← 1...N do
{
    if (i is divisible by 3) x ← x + i
}
print x
```

(a) Express the value of x as a function of N: and **(b)** What is x if $N = 302$.

a) ~~x = 3 + 6 + 9 + 12 + ...~~

$x = 5 + 3 + 6 + \ldots N/3$

$5 + \sum_{i=1}^{N/3} (3(i-1))$ $\quad i$ ✓ -0.5

b) $302/3 \rightarrow 300$

$5 + \sum_{i=1} (3(i-1)) = 134555$ ✓

Scanned with CamScanner

4. **[Marks 3]** 3/3
Solve the recurrence relation using the repeated substitution method,

$$T(n) = \begin{cases} 4T(n/3) + 2n, & n > 1 \\ c, & n = 1. \end{cases}$$

$\sum_{i=0}^{k-1} 4^k \times (2/3)^k n$

$\sum_{i=0}^{k-1} (4/3)^k 2n$ #

$\frac{4}{3} \times 2n$

$4T(n/3) + 2n$

$\Rightarrow 4\left[4T\left(\frac{n}{3^2}\right) + \frac{2n}{3}\right] + 2n = 4^2T\left(\frac{n}{3^2}\right) + \frac{8}{3}n + 2n$

$\Rightarrow 4^2\left[4T\left(\frac{n}{3^3}\right) + \frac{2n}{3^2}\right] + \frac{8}{3}n + 2n = 4^3T\left(\frac{n}{3^3}\right) + 4^2 \times \frac{2}{3^2}n + 4 \times \frac{2}{3}n + 2n$

$\Rightarrow 4^3\left[4T\left(\frac{n}{3^4}\right) + \frac{2n}{3^3}\right] + 4 \times \frac{2}{3}n + 4^2 \times \frac{2}{3^2}n + 2n$

$= 4^4T\left(\frac{n}{3^4}\right) + 4^3 \times \frac{2n}{3^3} + 4^2 \times \frac{2}{3^2}n + 4 \times \frac{2}{3}n + 2n$

$\vdots$

after k subs $\Rightarrow 4^k T\left(\frac{n}{3^k}\right) + \sum_{i=0}^{k-1} (4/3)^i 2n$

stop when $\Rightarrow \frac{n}{3^k} = 1 \Rightarrow n = 3^k \Rightarrow k = \log_3^n$

$\Rightarrow 4^{\log_3^n} T(1) + \frac{1 - (4/3)^k}{1 - 4/3} 2n$

$\Rightarrow T(n) \leq 4^{\log_3^n} + 2n$

$T(n) \in O(n)$

3/4

5. **[Marks 4]**
Solve the following recurrences using the Master theorem by giving tight Θ-notation bounds. Justify your answers. The Master theorem is: $T(n) = aT(n/b) + f(n)$,

$$\text{If } f(n) \in \Theta(n^d), \text{ where d} \geq 0, \text{ then } T(n) \in \begin{cases} \Theta(n^d), & \text{if } a < b^d, \\ \Theta(n^d \log n), & \text{if } a = b^d, \\ \Theta(n^{\log_b a}), & \text{if } a > b^d. \end{cases}$$

Another master theorem is given by $h(n) = f(n) / n^{\log_b a}$, then

$$T(n) \in \begin{cases} O(n^{\log_b a}), & \text{if } h(n) = O(n^r), r < 0, \\ \Theta(n^{\log_b a} \log^{k+1} n), & \text{if } h(n) = \Theta(\log^k n), k \geq 0, \\ O(f(n), & \text{if } h(n) = \Omega(n^r), r > 0. \end{cases}$$

(a) $T(n) = 4T(n/4) + 5\log^2(n)$

$a = 4$, $b = 4$, $f(n) = 5\log 2(n)$

$f(n) \ ?, \ n^{\log_4^4} = n$

$f(n) < n$

case 1 $\Rightarrow \emptyset(n)$

(b) $T(n) = 9T(n/3) + 3n^2$

$a = 9$, $b = 3$, $f(n) = 3n^2$

$f(n) \ ? \ n^{\log_3^9}$

$3n^2 = n^2$

case 2 $\Rightarrow \emptyset(n^2 \log n)$

(c) $T(n) = 7T(n/2) + n^3 \log n$

$a = 7$, $b = 2$, $f(n) = n^3 \log n$

$f(n) \ ? \ n^{\log_2^7}$

$n^3 \log n > n^{2,8}$

case 3 $\Rightarrow$ Finding c such that $af(n/b) \leq c f(n)$

$= 7 \frac{n^3 \log n}{2} \leq c \, n^3 \log n$

$- 1$

$= \frac{7}{2} \leq c < 1$

$c = 2$

6. **[Marks 5]** 2.5/5
Given an array A of size n. The array contains all *but* one of the integers from 0 to n. **(a)** Give the best algorithm you can to determine which number is missing if the array A is not sorted. **(b)** Analyze its asymptotic worst-case running time. For example, the list [3, 5, 0, 2, 1] is missing the number 4.

~~findmiss(A[0...n]){~~

~~for i←1 to n{ //sorting~~

a)

~~for~~

~~findmiss(A[0....n]){~~
~~x ← 0~~
~~for i←0..n{~~
~~x~~
~~if(x != A[i])~~
~~if (x != A[i]){~~
~~x++}~~
~~x++~~
~~}~~
~~return x~~
~~}~~

```
findmiss(A[0...n]){
  x ← 0
  for i←0..n{
    if(x != A[i])
      x++
    else
      return x
  }
}
```

-2.5

b) in worst case missing number in last index, so it will loop in entire array ⇒ O(n)